**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра МО ЭВМ**

отчет

**по лабораторной работе № 3**

**по дисциплине «Объектно-ориентированное программирование»**

**Тема: Контейнеры.**

|  |  |  |
| --- | --- | --- |
| Студентка гр.7303 |  | Дегтярева А.А |
| Преподаватель |  | Размочаева Н.В |

г. Санкт-Петербург

2019 г.

**Цель работы:**

Необходимо реализовать контейнеры вектор и список. Поведение реализованных функций должно быть таким же, как у классов std::vector и std::list.

**Ход работы:**

Для выполнения поставленной задачи были реализованы следующие структуры данных:

1. Были дописаны следующие методы для класса vector:
   1. explicit vector(size\_t count = 0) – создается вектор размера count;
   2. vector(InputIterator first, InputIterator last) – создается вектор со значениями от first до last;
   3. vector(std::initializer\_list <Type> init) – создается вектор с заданным диапазоном значений;
   4. vector(const vector& other) – конструктор копирования;
   5. vector(vector&& other) – конструктор перемещения;
   6. ~vector() – деструктор;
   7. vector& operator=(const vector& other) – оператор присваивания копирования;
   8. vector& operator=(vector&& other) – оператор присваивания перемещения;
   9. void assign(InputIterator first, InputIterator last) – копирование в вектор значений от first до last;
   10. void resize(size\_t count) – изменение размера вектора;
   11. iterator erase(const\_iterator pos) – удаление элемента в заданной позиции;
   12. iterator erase(const\_iterator first, const\_iterator last) – удаление элементов от first до last;
   13. iterator insert(const\_iterator pos, const Type& value) – вставка элемента в заданную позицию;
   14. iterator insert(const\_iterator pos, InputIterator first, InputIterator last) – вставка нескольких элементов в заданную позицию;
   15. void push\_back(const value\_type& value) – добавление элемента с заданным значением в конец;
2. Были дописаны следующие методы для класса list:
   1. void push\_back(const value\_type& value) – вставка элемента в конец списка;
   2. void push\_front(const value\_type& value) – вставка элемента в начало списка;
   3. reference front() – получение значения первого элемента;
   4. const\_reference front() const – получение постоянного значения первого элемента;
   5. reference back() – получение значения последнего элемента;
   6. const\_reference back() const – получение постоянного значения последнего элемента;
   7. void pop\_front() – извлечение первого элемента списка;
   8. void pop\_back() – извлечение последнего элемента списка;
   9. void clear() – отчистка списка;
   10. bool empty() const – проверка на пустой список;
   11. size\_t size() const – получение размера списка;
   12. ~list() – деструктор;
   13. list(const list& other) – конструктор копирования;
   14. list(list&& other) – конструктор перемещения;
   15. list& operator= (const list& other) – оператор присваивания;
   16. iterator insert(iterator pos, const Type& value) - вставка элемента в заданную позицию;
   17. iterator erase(iterator pos) – удаление элемента из заданной позиции;
3. Были дописаны следующие методы для класса list\_iterator:
   1. list\_iterator& operator = (const list\_iterator& other) – оператор присваивания;
   2. bool operator == (const list\_iterator& other) const – проверка на равенство двух итераторов;
   3. bool operator != (const list\_iterator& other) const – проверка на неравенство двух итераторов;
   4. reference operator \* () – разыменование;
   5. pointer operator -> () – получение адреса элемента;
   6. list\_iterator& operator ++ () – постфиксный переход к следующему итератору;
   7. list\_iterator operator ++ (int) – преффиксный переход к следующему итератору.

**Заключение**

В ходе выполнения данной лабораторной работы была изучена тема контейнеры. Были реализованы методы для контейнеров вектор и список.